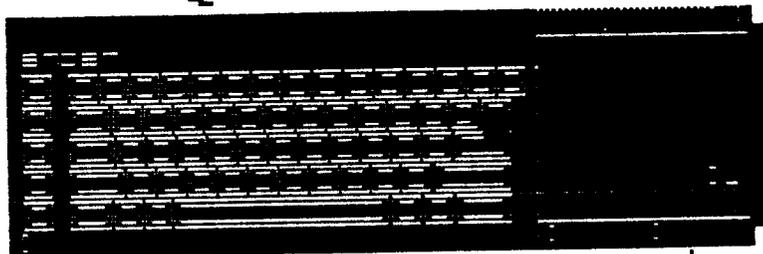


DEC.  
JAN.  
1990

# QL\_DOC

#13

\$1.50



## Le Bulletin Québécois pour les Utilisateurs Sinclair QL

QL\_DOC est publié à tous les 2 mois et est réalisé avec l'aide du logiciel PAGE DESIGNER II avec OMNI-DUMP de SECTOR SOFTWARE. L'impression se fait lentement mais sûrement sur une imprimante EPSON LX-800 (la STAR NX1000 étant en réparation).

Vos articles, commentaires, suggestions sont appréciés.

Rédacteur: Réal Gagnon, 8286 St-Hubert, Montréal (Québec)  
CANADA H2P 1Z3 (514) 381-6462

Abonnement 1 an (6 nos) : \$12.00

### SOMMAIRE

FORMAT_BIN	_____	R. Gagnon
DSKNAME_BAS	_____	R. Gagnon
Essai de ACT de DP	_____	S. Vandezande
Les Langages	_____	U. Harmer
ABACUS APPLICATIONS	_____	R. Gagnon

QL\_DOC souhaite un JOYEUX NOEL  
et BONNE & HEUREUSE ANNEE à  
tous les QListes!

# SINCLAIR NEWS

## ATARI QL.....

Il y a Tony Tebby, le papa du QDOS, qui répète à qui veut l'entendre qu'il va développer une version spécifique du QDOS pour l'Atari ST. Ce QDOS se présentera sous la forme d'un ROM et sera un remplacement direct du TOS, le système d'exploitation actuelle de l'ATARI. Cela veut dire qu'un ST ainsi équipé deviendrait un SUPER QL! Les programmes du QL fonctionneront sans problème et BEAUCOUP plus rapidement grâce au vrai 68000 du ST et au QDOS optimisé pour le matériel ATARI. A suivre...

## MINERVA OS pour le QL.....

Comme si le QDOS nous suffisait plus, voici qu'un système d'exploitation compatible fait son apparition. Il s'agit du MINERVA OS, un remplacement direct du QDOS à l'intérieur de la machine. Les 2 principaux avantages sont la vitesse d'exécution augmentée et la correction des BUGs du QDOS original. MINERVA OS est un produit d'une compagnie britannique ayant pour nom QUIEV.

## Un QL XT.....

La compagnie ouest-allemande ABC ELECTRONIC introduira sous peu un contrôleur de disque dur pour le QL. La particularité de l'interface est qu'elle ajoute une fente d'expansion pour une carte compatible PC (n'importe quoi sauf carte vidéo ou carte mémoire). Avec le contrôleur de disque dur, il y a un toolkit pour la gestion des sous-répertoires. Il sera possible d'utiliser tout disque dur compatible avec la norme ST506, c'est à dire ce qui il y a de plus répandu. Le prix de l'interface et du contrôleur est de 129 livres sterling.

ABC ELEKTRONIC, Hügelsstr. 10-12, 4800 Bielefeld 1, W. Germany  
fax: 0521-892615

## ERRATA . . .

Dans le numéro 12, l'article intitulé ABACUS APPLICATIONS contenait les erreurs suivantes:

Lors du classement en 4 colonnes voici les 3 commandes COPY à faire:

F3-C-A0:A12(ENTER)-B3

F3-C-A13:A17(ENTER)-C3

F3-C-A18:A22(ENTER)-D3

Une fois ceci fait, nous faisons la commande: RUBOUT A0:A22

Dans le programme BASIC ABA01R\_BAS, on peut rajouter la ligne suivante:

405 DELETE medium%\$'direct\_tap'

# SINCLAIR NEWS

DES NOUVELLES du VIEUX CONTINENT.....

M. J-L Dianoux, notre correspondant du club QLCF, communique à QL\_DOC les derniers développements QLiens en Europe.

Le 21 octobre 1989 avait lieu à Bruxelles, le MICROFAIR européen. Il y avait pas mal de monde, 560 personnes au moins ayant été pointées. Le plus important de tout était bien sûr la présence, outre des clubs européens les plus gros (anglais et allemands), de Tony Tebby, le concepteur du QDOS, qui continue opiniâtement mais avec beaucoup d'espoir la mise au point du SMS 2 (Small Microcomputer operating System 2), en le traduisant au fur et à mesure pour les processeurs INTEL... Le portage de ce système d'exploitation se poursuit comme prévu sur ATARI. QRAM 2 était en démonstration à l'état de prototype, c'est devenu un produit extrêmement complet et puissant.

Quant au THOR XVI, son évolution nous laisse extrêmement perplexes et attentistes: la dernière version (6.4) était horriblement boguée, nul ne sait combien et à qui DANSOFT les vend (cher, d'ailleurs)...

Les disques durs de MIRACLE SYSTEMS fonctionnent très bien. Ceux de REBEL ELECTRONICS ont eu des difficultés de fabrication, mais les premiers exemplaires ont été livrés. ABC ELECTRONIKS présentait la sienne, mais nous méfions d'eux au niveau qualité. THE PAINTER est un très joli logiciel de dessin, qui exploite parfaitement l'interfaçage POINTER de QJUMP.

Une émulation du QL sur AMIGA existe et fonctionne fort bien, hormis certaines limitations temporaires: il s'agit d'une disquette que l'on charge, et l'AMIGA est transformé en QL! C'est du domaine public. Pour plus d'infos, on écrit à UGA SOFTWARE, Ron Fontaine, CROSENSTEIN 33 14, 3704 NH ZEIST, PAYS BAS. Attention: dans la version actuelle, les disquettes écrites par l'AMIGA émulé QL ne peuvent être lues par un QL.

## BBS SINCLAIR à Ottawa.....

Le club OTTAWA-HULL offre maintenant un service de BBS gratuit à tous les utilisateurs SINCLAIR canadiens. Pour le moment, le système est limité à la vitesse 300 bauds mais devrait adopter le 1200 bauds bientôt.

Lors du premier appel, à la question NAME?, on répond SPECIAL GUEST et le mot de passe est CLIVE. Lorsque vous quitterez, laissez un message au SYSOP pour lui signifier que vous voulez vous inscrire en donnant votre nom, adresse et numéro de téléphone, plus un mot de passe personnel.

Comme le BBS se trouve à Ottawa, il y a des frais de longue distance. Il faut donc appeler durant la soirée ou durant la fin de semaine. Le BBS est en marche 7 jours sur 7, 24 heures par jour.

OTTAWA-HULL SINCLAIR BBS (613) 745-8838 300 bauds, 8bits, pas de parité

# Nouveau FORMAT

La commande FORMAT est très dangereuse, elle commence immédiatement son travail sans nous avertir qu'elle va tout effacé sur son passage. Parce que j'ai déjà perdu des trucs en formatant la mauvaise cartouche, j'ai écrit ce petit programme qui redéfinit la commande FORMAT pour la rendre plus sécuritaire. En effet, la nouvelle syntaxe de la commande FORMAT nous oblige à mettre des guillemets maintenant; FORMAT mdu\_ n'est plus valide; FORMAT "mdv\_" est acceptable. Si le paramètre est accepté, un message nous avertissant du danger imminent apparaît et on nous demande d'appuyer sur la touche F1 pour débiter le formatage ou bien de faire F2 pour annuler la procédure. En réalité, seule la touche F1 fera débiter le formatage, toutes les autres annuleront l'opération.

On charge le programme de cette façon:

```
a= RESPR(256)
LBYTES xxxx_FORMAT_BIN,a
CALL a
```

ou xxxx est le support qui contient le fichier FORMAT\_BIN. A partir de ce moment, la nouvelle syntaxe de la commande FORMAT est en vigueur.

Le canal 80 se doit d'être ouvert, si il est fermé, on ne verra pas les indications à l'écran. FORMAT\_BIN fonctionne sur des QLs équipé avec la ROM JS ou JSU.

Il y a 3 façons d'obtenir ce programme.

1- Envoyer 4\$ plus le support (mdv/5.25"/3.5") sur lequel vous voulez recevoir FORMAT\_BIN à QL\_DOC.

2- Taper le programme SuperBASIC suivant qui va créer le fichier FORMAT\_BIN.

3- Taper le programme source assembleur et compiler-le avec un assembleur quelconque; Metacomco MASM par exemple.

> PROGRAMME SuperBASIC POUR CREER FORMAT BIN

```
100 REMark CHARGEUR_BAS
110 REMark pour QL_DOC
120 :
130 INPUT "Où sauver les codes (ex. flpl_)?";sauve$
140 IF LEN(sauve$)=0:sauve$="flpl_"
150 PRINT "Un instant s.v.p."
160 :
170 RESTORE 1000
180 memoire=RESPR(1000);base_memoire=memoire
190 compte=0
```

```

170 RESTORE 1000
180 memoire=RESPR(1000):base_memoire=memoire
190 compte=0
200 :
210 REPEAT boucle
220 IF EOF:EXIT boucle
230 READ octet:POKE memoire,octet:memoire=memoire+1
240 compte=compte+1
250 PRINT ".";
260 END REPEAT boucle
270 :
280 DELETE sauve%format_bin"
290 SBYTES sauve%format_bin",base_memoire,compte
300 PRINT "Terminé!"
1000 DATA 67, 250, 0, 10, 52, 120, 1, 16, 78, 146
1001 DATA 78, 117, 0, 1, 0, 16, 6, 70, 79, 02
1002 DATA 77, 65, 84, 0, 0, 0, 0, 0, 0, 0
1003 DATA 52, 120, 1, 22, 78, 146, 74, 128, 182, 86
1004 DATA 112, 241, 182, 124, 0, 1, 102, 78, 211, 206
1005 DATA 47, 9, 145, 200, 54, 120, 0, 200, 78, 147
1006 DATA 67, 250, 0, 78, 78, 147, 112, 0, 78, 65
1007 DATA 54, 40, 0, 138, 56, 40, 0, 138, 184, 67
1008 DATA 103, 248, 184, 124, 0, 232, 32, 95, 102, 36
1009 DATA 112, 3, 78, 66, 74, 120, 102, 40, 47, 2
1010 DATA 145, 200, 54, 120, 0, 206, 78, 147, 67, 250
1011 DATA 0, 84, 54, 120, 0, 208, 78, 147, 34, 31
1012 DATA 54, 120, 0, 206, 78, 147, 145, 200, 67, 250
1013 DATA 0, 104, 54, 120, 0, 208, 78, 147, 78, 117
1014 DATA 0, 52, 32, 115, 101, 114, 97, 32, 78, 79
1015 DATA 82, 77, 65, 84, 69, 32, 101, 116, 32, 69
1016 DATA 78, 78, 65, 67, 69, 33, 32, 10, 32, 32
1017 DATA 32, 78, 49, 46, 46, 46, 32, 79, 117, 105
1018 DATA 32, 32, 32, 78, 58, 46, 46, 46, 32, 78
1019 DATA 111, 110, 32, 18, 0, 38, 32, 115, 101, 99
1020 DATA 116, 101, 117, 114, 115, 32, 66, 79, 78, 83
1021 DATA 32, 115, 117, 114, 32, 117, 110, 101, 32, 112
1022 DATA 111, 115, 115, 105, 98, 105, 108, 105, 116, 131
1023 DATA 32, 108, 101, 32, 0, 11, 10, 84, 101, 114
1024 DATA 109, 105, 110, 131, 46, 32, 10, 0

```

```

** Liste du programme assembleur de FORMAT_BIN
** CE PROGRAMME APPARTIENT AU DOMAINE PUBLIC
**

```

```

*****
*   FORMAT alternatif v.1   *
*                           *
*   novembre 1989         *
*   par Réal Gagnon      *
*   Montréal, Québec    *
*****

```

```

ERR_BP   equ   -15      erreur BAD PARAMETER
ID_FORMAT equ    3      routine GDOS format

```

BP_INIT	equ	\$110	"	GDOS initialisation extension
CR_GTSTR	equ	\$116	"	GDOS passage de paramètre (string)
UT_MTEXT	equ	\$D0	"	GDOS imprime message à l'écran
UT_MINT	equ	\$CE	"	GDOS imprime un entier à l'écran
NT_INF	equ	0	"	GDOS renvoie la base des Sys.Var.
SU_ARBUF	equ	\$A9		Sys.Var. indiquent la dernière touche pressée
DEFPROC	lea.l	PROC,a1		initialisation de la nouvelle commande FORMAT
	move.u	BP_INIT,a2		
	jsr	(a2)		
	rts			
PROC	dc.u	1		nombre de procédure à définir
	dc.u	FORMAT-t		
	dc.b	6		
	dc.b	'FORMAT'		
	dc.u	0		fin des procédures
	dc.u	0		nombre de fonction à définir
	dc.u	0		fin des fonctions
FORMAT	move.u	CR_GTSTR,a2		va chercher paramètre
	jsr	(a2)		
	tst.l	d0		
	bne.s	fin		sortie si erreur
	moveq	#ERR_BP,d0		
	cmp.u	#1,d3		vérifie si il y a un seul paramètre
	bne.s	fin		si non sortie avec BAD PARAMETER
	adda.l	a6,a1		
	move.l	a1,-(sp)		sauve R1 (dessus de pile)
*				Ici (R6,R1 qui sauve sur la pile) contient
*				
*				le nom du périphérique sous la forme:
*				2 octets=longueur
*				le reste=le nom du "device" à FORMATER.
ATTENTION	suba.l	a0,a0		imprime un message à l'écran (CANFLA0)
	move.u	UT_MTEXT,a3		impression du "device" (R1)
	jsr	(a3)		
	lea.l	MESSAGE1,a1		
	jsr	(a3)		
	moveq	NT_INF,d0		
	trap	#1		R0=base des systèmes variables
	move.u	SU_ARBUF(a0),d3		
BOUCLE	move.u	SU_ARBUF(a0),d4		attend une nouvelle touche
	cmp.u	d3,d4		
	beq.s	BOUCLE		
	cmp.u	#5E8,d4		nouvelle touche = F1?
FMT_DSK	move.l	(sp)+,a0		R0 pointe vers la pile=nom du "device"
	bne.s	FIN		
	moveq	#10_FORMAT,d0		routine GDOS FORMAT
	trap	#2		
	tst.l	d0		
	bne.s	DEPART		

```

*
*
      move.l d2,-(sp)
      suba.l a0,a0
      move.u UT_MINT,a3      l'expression du nombre de secteur OK
      jsr    (a3)
      lea.l  MESSAGE2,a1
      move.u UT_MTEXT,a3
      jsr    (a3)
      move.l (sp)+,d1      sur une possibilité de ...
      move.u UT_MINT,a3
      jsr    (a3)
FIN   suba.l a0,a0
      lea.l  OK,a1         message de sortie
      move.u UT_MTEXT,a3
      jsr    (a3)
DEPART rts

MESSAGE1 dc.u  $34
          dc.b  ' sera FORMATE et EFFACE! ',a0
          dc.b  ' F1... Oui F2... Non ',a0
MESSAGE2 dc.u  $26
          dc.b  ' secteurs BONS sur une possibilité de '
OK        dc.u  $0B
          dc.b  $0A,'Terminé. ',a0
          end

```

# LANGAGES

de U. Harber, publié dans TS BULLETIN OTTAWA-HULL UC  
traduction et adaptation de QL\_DOC

Nous pouvons nous demander, "Pourquoi ne pouvons-nous pas communiquer avec un ordinateur simplement en anglais (ou en français) et utiliser celui-ci comme langage informatique?".

Les premiers ordinateurs n'avaient ni la vitesse ni la mémoire pour analyser une phrase en anglais et encore moins fournir un simple réponse à une question. Au début, les langages informatiques étaient juste capables de manipuler quelques formules mathématiques et fournir des réponses à des équations scientifiques qui auraient nécessité de longues heures de programmation en langage machine. Le langage FORTRAN (FORmula TRANslation), inventé par J. Backus de IBM dans le milieu des années 50, réalisait très bien cette fonction et est encore utilisé aujourd'hui par les ingénieurs et scientifiques.

Plus tard, le monde des affaires était de plus en plus envahis par les minis ordinateurs IBM et une autre tentative de produire un langage plus facile fut faite avec l'introduction du langage COBOL (COMmon Business Oriented Language). Quoique spécifiquement créé pour l'administration et la comptabilité, COBOL se voulait un langage d'usage général.

Avec une popularité grandissante, de nouveaux langages firent leur apparition pour faciliter l'enseignement de l'informatique à l'université. De ce groupe, on remarque le BASIC (Beginner's All-purpose Symbolic Instruction Code) et le Pascal (nommé ainsi d'après le mathématicien-philosophe français Blaise Pascal). Ces deux langages ont trouvé plusieurs applications à l'extérieur de l'enseignement laissant ainsi dans l'oubli leur origine pédagogique.

Les tentatives de créer des langages produisant des programmes plus rapide d'exécution nous ont donné le langage C. Ce nom plutôt bizarre nous rappelle que le "C" est le résultat de 2 tentatives précédentes qui ont donné le langage "A" et le "B" (appelé aussi BCPL). Le créateur du Pascal, le Professeur Nicholas Wirth propose maintenant un langage appelé Modula (en fait Modula-2 est la dernière version de ce langage). Modula combine la vitesse d'exécution du C mais est plus facile à maîtriser.

L'intelligence artificielle est une caractéristique du langage LISP (LIST Processing) et du PROLOG. Séduisante au début des années 80, l'intelligence artificielle excite beaucoup moins de monde aujourd'hui.

A un moment donné, l'armée américaine décide de standardiser les langages de programmation de tous ses systèmes informatiques. Pour ce faire un nouveau langage sera créé et il aura pour nom ADA.

FORTH se voulait être le langage de la 4ème génération mais il n'a jamais vraiment décollé peut-être à cause de la notation polonaise inversée...

PL/1 (Programming Language) est utilisé quelque fois sur les minis ordinateurs, il ressemble au BASIC mais en plus compliqué. APL (A Programming Language) est surtout utilisé pour traiter de larges matrices de chiffres mais est très bizarre car il utilise des symboles et non pas des lettres et demande l'utilisation de claviers spéciaux.

La tendance d'aujourd'hui est dans l'utilisation de langage spécifique à une application ou bien un logiciel qui offre son propre langage de programmation. Prenons par exemple, DBASE pour les base de données et LOTUS 123 pour les tableurs dans le monde MS-DOS (les équivalents QLiens sont ARCHIVE et ABACUS).

**BACUS APPLICATIONS**

Si vous avez besoin de calculer des heures avec ABACUS, il est plus facile de travailler avec des heures et des minutes qu'avec des heures et des fractions (1 heure et 25 minutes au lieu de 1.4167 heures).

Prenons une feuille de calcul qui ressemble à ceci :

	A	B	C
1	Hre		Min
2	-----		
3	2		52
4	3		15
5	1		10
6	5		48
7	7		35
8	-----		
9	20		40

Pour obtenir les totaux apparaissant à la ligne 9 nous avons besoin des 2 formules suivantes :

cellule A9:  $SUM(A3:A7)+INT(SUM(C3:C7)/60)$

cellule C9:  $SUM(C3:C7)-(60*INT(SUM(C3:C7)/60))$

La formule dans A9 additionne les heures, puis les minutes sont additionnées pour être divisées par 60.

La formule dans C9 additionne les minutes, divise par 60 nous montre le reste. En fait si ABACUS avait une fonction MOD qui retournerait le reste d'une division de 2 entiers, la formule serait plus simple, par exemple on aurait  $MOD(SUM(C3:C7),60)$  pour obtenir le reste de la division de la somme des minutes par 60. Mais comme la fonction MOD n'est disponible avec ABACUS on se débrouille comme on peut avec des équivalents.



Vous est-il déjà arrivé d'avoir subitement l'envie folle d'écrire vos propres jeux d'aventure, mais de ne pas vouloir pas vous fatiguer à écrire un programme en Basic, Pascal ou autre Assembleur ? Si oui, voici la solution : The Adventure Creation Tool de Digital Precision.

Ce programme d'aide à la conception de jeux d'aventures est en fait une version améliorée du programme APT (Adventure Programming Tool) qui est sorti chez Shadow Games en automne 1987. ACT est disponible uniquement sur disquette se compose de 2 grandes parties : le programme 'Captain' et les divers éditeurs.

Détailons d'abord les éditeurs. Premièrement, nous avons le GRAPHICS DESIGNER (GDES\_task). Ce programme nous permet de dessiner les écrans que l'on trouvera dans notre jeu. Il propose de nombreuses options mais celles-ci sont malheureusement très peu 'user-friendly'. Notons entre-autres : Lignes, Arcs de cercles, Ellipses, Boîtes, Triangles, 2 modes de remplissage des figures,... Il est d'ailleurs possible de dessiner des écrans en haute et en basse résolution. Lorsque nous chargeons le programme, il nous demande de situer une fenêtre à l'écran. Cette fenêtre est celle dans laquelle notre chef-d'oeuvre sera affiché. Ensuite nous entrons la résolution d'écran et nous voici devant une feuille blanche (noire, en réalité). Pour accéder aux commandes, on pousse la touche qui s'y rapporte. Si vous avez un trou de mémoire, vous pouvez toujours presser 'F1' pour obtenir la liste des options disponibles et les touches par lesquelles on les obtient. On revient à notre feuille en poussant une deuxième fois sur F1. Voilà pour ce qui est de l'éditeur de graphiques.

Passons maintenant à l'éditeur de sons. Ce programme est destiné à définir les bruitages et morceaux de musique que l'on trouvera dans le jeu. La partition de sons se présente sous forme d'une fenêtre qui contient les paramètres des sons (longueur, hauteur,...) et leurs coordonnées. Ces coordonnées sont très intéressantes car l'on peut passer d'un son à l'autre comme, en Basic, on passe d'une ligne à une autre par l'instruction 'GO TO'.

Ensuite, il reste les éditeurs de vocabulaire (UOCedt\_task), de localisations (LOCedt\_task), et de messages (MSGedt\_task). Pour élaborer un (bon) jeu d'aventure, il faut donc faire un plan sur une feuille de papier car il faut bien se rendre compte que le programme ne fera pas le scénario à votre place.

Ensuite, on rentre le vocabulaire, les messages, la position de pièces, les graphiques, le son et c'est là qu'intervient le programme 'CAPTAIN'. Il s'agit en fait d'un programme 'linker' qui relie les différentes parties du jeu en un seul programme. Notons au passage qu'il est possible d'afficher simultanément les graphiques et les textes du jeu en haute et en basse résolution à l'écran, ce qui très utile si l'on veut, par exemple, avoir 80 colonnes disponibles pour le texte pendant que l'on dispose de 8 couleurs pour le graphisme.

Sur la disquette d'ACT, nous trouvons aussi 2 jeux comme exemple de ce que l'on est capable d'obtenir grâce à ce programme. Le premier est une mini-aventure avec les fichiers sources afin que l'on puisse, en guise d'apprentissage, trifouiller un peu dans un programme existant. Le deuxième programme s'appelle 'Imagine' et, personnellement, je trouve que c'est un des meilleurs jeux d'aventures disponibles sur le QL.

En conclusion, je dirai que ACT est un très bon programme pour celui qui désire réaliser des jeux d'aventures. Mais le principal défaut de ce programme, c'est qu'il demande un apprentissage assez long et ne sera donc vraiment intéressant que si l'on a le projet d'élaborer au minimum 2 ou 3 jeux et de les commercialiser par la suite.

S. Vandezande

ACT Special Edition  
49.95 Livres Sterling  
DIGITAL PRECISION  
222, The Avenue  
London, E4 9SE  
Angleterre



‡ Impression avec ABACUS

Il est relativement facile de contrôler l'impression d'une feuille de calcul avec la fonction CHR en utilisant cette dernière pour l'envoi de codes de contrôle à l'imprimante.

Si nous faisons CHR(64), un "A" apparaît à l'écran. Si on donne un argument inférieur à 32 à CHR alors on ne verra rien à l'écran ni à l'imprimante. Il suffit de mettre en premier un CHR(0) pour que le caractère de contrôle soit dirigé vers l'imprimante.

Par exemple, CHR(0)+CHR(15) mettra le mode COMPRESSE de l'imprimante en marche et CHR(0)+CHR(18) le mettra hors fonction. Le mode compressé est intéressant car il permet d'obtenir jusqu'à 132 caractères sur un seul ligne. On pourrait en avoir encore plus, 160 caractères par ligne si votre imprimante supporte le mode COMPRESSE ELITE avec la commande CHR(0)+CHR(15)+CHR(27)+"M". Cependant si on se met en mode COMPRESSE sur l'imprimante, on doit le dire à ABACUS que l'on veut plus que 80 caractères par ligne. Pour ce faire on va à l'option DESIGN, PRINTER PAPER WIDTH et on donne la bonne valeur.

Le soulignement est maintenant facilement accessible, entrer ceci dans A1 et imprimer le tout.

```
CHR(0)+CHR(27)+"-1"+"QL_DOC youppidou"  
et dans une cellule DIFFERENTE CHR(0)+CHR(27)+"-0"  
pour mettre le mode soulignement hors-fonction.
```

Il est à noter que si le BORDER est imprimé, la partie du haut ne sera pas affectée par les codes présents dans notre feuille de calcul. Le CHR(0) doit être le premier item dans la cellule.

## Au nom de la DISQUETTE

Lors du formatage d'une disquette, nous avons la possibilité de donner un nom à celle-ci. Cependant il n'est pas possible de renommer une disquette une fois formatée à moins bien sûr de procéder à un reformatage qui effacera en même temps la disquette.

Le programme SuperBASIC suivant vise à régler ce problème. Il sera désormais possible de renommer une disquette sans être obligé de reformater celle-ci.

À l'exécution du programme, on demande le numéro du lecteur de disquette à être renommé. Il est à noter que ce programme ne fonctionne qu'avec des disquettes et qu'il n'est pas possible de renommer des microdrives par exemple. Le nom actuel de la disquette est alors affiché et on demande le nouveau nom. Nous sommes limité à une longueur de 10 caractères.

```
100 REMARK DISKRENAME v1  
110 REMARK par Réal Gagnon  
120 :  
130 REPEAT loop  
140 PRINT#0,"Disquette no.!!";  
145 d$=INKEY$(←1)  
150 IF d$ INSTR '123456789':PRINT#0,d$:EXIT loop  
160 END REPEAT loop  
165 :  
170 OPEN#10,"f\p"&d$&"_d020"  
180 :  
190 GET#10\1,d$  
200 PRINT#0,"Nom actuel :!"&d$(5 TO 14)  
210 INPUT#0,"Nouveau nom :!"&n$  
220 LET a$(5 TO 15)=n$  
230 PUT#10\1,a$  
240 :  
250 CLOSE#10
```